Vol.2009-NL-193 No.3
2009/9/28

Web 文書中の単語クリックログの解析から未知単語を予測する語義注釈システム

江原 遥[†1] 二宮 崇[†1] 中川 裕志[†1]

近年，第二言語で書かれた Web ページの読解のために，語義注釈システムを用いて Web ページを読むユーザが増えている．語義注釈システムを用いると，ユーザは，読解を妨げるユーザの知らない語（非既知語）に遭遇した場合，クリックなどの操作により語義を表示させ，語の意味を知ることができる．しかしながら，語義注釈システムのログである “単語クリックログ” はこれまで活用されてこなかった．本研究では，単語クリックログを解析することにより，読解の障害となる非既知語を予測し，ページを表示する際に予め語義注釈を付与することにより読解を容易にするシステムを提案する．予測手法は，TOEFL などの言語テストで使用されている項目反応理論の基礎である Rasch モデルを用いた．予測精度を向上させるため，Rasch モデルに素性を追加して拡張した．高いスケーラビリティと可用性を実現するため，クラウド環境である Google App Engine 上でシステムを実装した．高いスケーラビリティと即応性を実現するため，予測手法には逐次学習法である Stochastic Gradient Descent を用いた．実験によって，これらの手法の効果を確認した．

A Predictor of Words Unknown to Users by Analyzing Clicked Word Logs

Yo Ehara,[†1] Takashi Ninonimiya[†1] and Hiroshi Nakagawa[†1]

Recently, more and more users are now browsing Web pages written in second languages.When users browse those Web pages, unfamiliar words can be obstacle for reading. Word glossing systems help users to read the words that are unfamiliar to the user by displaying the meaning of the unfamiliar word, and a user is able to know the meaning of it. The system displays its meaning in a pop-up window when a user selects a word by clicking on it or mousing over it. These system accumulates the “clicked word log”. These logs are valuable to support reading although previous glossing systems do not utilize them. This paper proposes a novel glossing system that analyzes clicked word logs and predicts the words unfamiliar to the user and glosses the words so that they would not prevent the user from reading. Item response theory (IRT) is employed for the prediction because it is widely used in language testing, such as TOEFL. We extended IRT by adding features so that it can achieve higher accuracy. We also designed our system to be scalable.Google App Engine, a cloud environment for Web applications, and stochastic gradient descent, an online algorithm, are also employed for the scalability. We evaluated our system by experiments on prediction accuracy.

1. はじめに

近年，情報のグローバル化に伴い，英語をはじめとする第二言語で書かれた Web ページを読むニーズが増えている．第二言語で書かれた Web ページを読む際には，ユーザが知らない語（非既知語）が読解を妨げる原因の一つとなる．この問題に対応するため，“語義注釈システム” が提案されてきた．語義注釈システムを用いると，ユーザは，知らない語（非既知語）に遭遇した場合，クリックまたはマウスオーバーなどの語の選択操作により，語義を表示させ，語の意味を知ることができる．**図 1** に挙げる pop 辞書[1)] では，マウスオーバーした非既知語の語義をポップアップで表示している．また，ドラッグ操作で選択した非既知語の語義を Web ページ中に埋め込むシステムも提案されている[2)]．

語義注釈システムでは，ユーザがクリックした単語を記録することにより，ユーザの非既知語のログが蓄積される．このログを，本稿では “単語クリックログ” と呼ぶ．単語クリックログは，読解の障害となる非既知語のリストであるので，読解支援にとって有用な情報であると考えられる．既存の語義注釈システムでは，単語クリックログは活用されてこなかったが，単語クリックログを解析することにより，読解の障害となる非既知語を予測し，予め語義を付与して読解を容易にすることが可能となると考えられる．

図 1 pop 辞書[1)] での注釈の例

†1 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo

本研究では，単語クリックログを解析し，既存の語義注釈システムに対しユーザの語彙を予測する機能を付加したシステムを提案する．システムは，http://socialdict.appspot.com/ にて運用されている．本システムは，非既知語を自動的に予測し，その語に語義の注釈を付与する．ユーザが本システムにログインし，本システムを通して Web ページを閲覧した図が**図 2** である．赤く着色された部分が非既知と判別された部分であり，語義注釈が付与されている．黄色く着色された部分が既知と判別された部分である．

本システムの予測機能により，クリックすることが不可能な印刷媒体やクリックすることが難しいモバイル端末でも，注釈を付与することが可能となる．利用者が未読のテキストに対して，予め単語リストを作成することも可能となる．また，本システムは判別の際に，“ユーザの語彙力” を推定する．今まで Web アプリケーションからは利用が難しかった情報であるため，さまざまな応用が想定される．例えば，Web ページの集合をユーザにとって読みやすい順番に並び替えたりすることなどが挙げられる．

提案するシステムは，より多くのユーザからの単語クリックログを収集することができれば，より高精度な予測が可能となると考えられる．より多くのユーザからの単語クリックログを収集するためには，ユーザ数が増加しても対処できるように，スケーラビリティ，即応性，可用性のある設計が求められる．

本論文の構成について述べる．§2 では，提案するシステムの構造について述べる．§3 では，予測に用いたアルゴリズムについて述べる．§4 では，予測精度の実験について述べる．§5 で，結論と今後の課題について述べる．

2. システムの構造

本節では，提案する語義注釈システムの構造について説明する．**図 3** に提案するシステムの構造を図示する．

(0) ユーザはユーザ識別子 u を本システム に渡す．既存のシステムでは，ユーザ適応をしないため，この動作は不要であった．

The easing of border restrictions, to begin Friday, means more South Korean citizens and cargo lorries(貨物自動車,トラック,トロッコ) will be allowed to travel to Kaesong, which employs mostly North Korean workers in Southern-owned businesses.

図 2 提案するシステムでの注釈の例

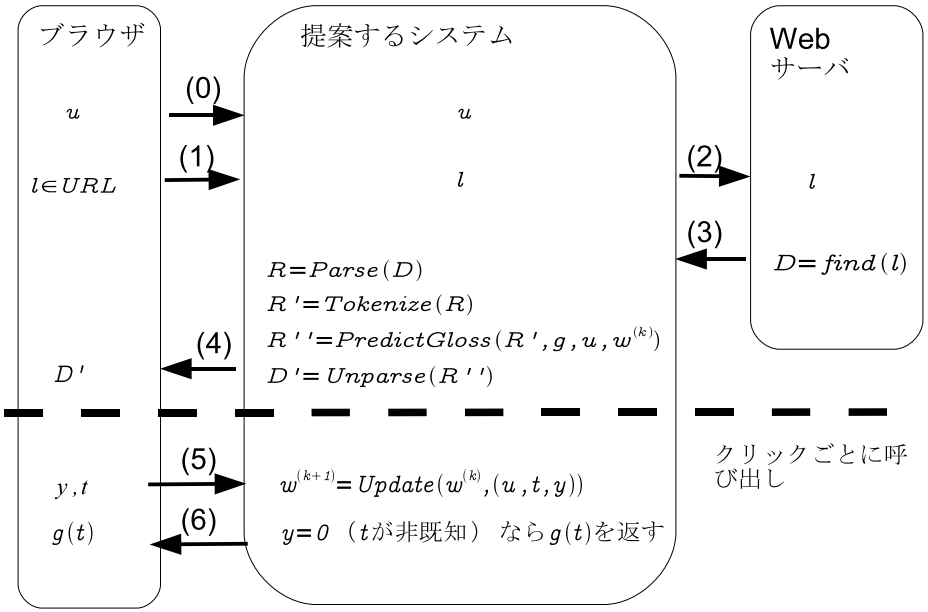

図 3 提案するシステムの構造

(1) ユーザは，ブラウザを通じて $l \in URL$ を本システム に渡す．URL は URL の集合とする．

(2) 本システム は，渡された l が指し示す “Web サーバ” にアクセスする．

(3) Web サーバは l を受け取り，l に対応する Web ページ D を探索し（$D = find(l)$），本システムに返す．

(4) 本システム は，D に注釈をつけて返す．この処理については本文を参照．

図 3 (4) における Web ページ D の例を，**図 4** に示す．“Voters cast their ballots.” という文章が，“⟨html⟩” や “⟨body⟩” といったタグで囲まれて整形されている．このような Web ページは，**図 5** (a) に示すように，テキストを葉，葉以外のノードをタグとするような木構造で表現することが可能である．全ての Web ページの集合を Dom_D，全ての木構造の集合を Dom_T と表す．Web ページ $D \in Dom_D$ を受け取り，木構造 $R' \in Dom_T$ を返す処理を $R' = Parse(D)$ と書く．逆に木構造 $R'' \in Dom_T$ を受け取り Web ページ $D' \in Dom_D$ を返す処理を $D' = Unparse(R'')$ と書く．

```
<html><body>
Voters cast<br>their ballots.
</body></html>
```

図 4 Web ページ D の例

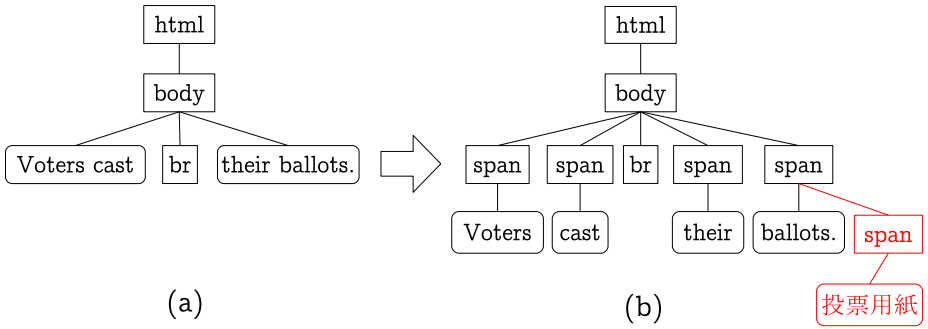

図 5 (a) 図 4 の例の木構造, (b) ブラウザに返却される木構造．赤い部分が付与される注釈の例．

図 3(4) では，図 5(a) に対して，“トークン化” と “予測機能による注釈” を行い，図 5(b) のように変換する．これらを，それぞれ “Tokenize” と “PredictGloss” という 2 つの木構造を取り木構造を返す関数で表現する．“Tokenize” は木構造 $R \in Dom_T$ を受け取り，R の葉であるテキストをトークン化した木 $R' \in Dom_T$ を返す．図 5(a) をトークン化したものが，図 5(b) の赤字部分を除いた木構造である．図 5(b) 中でトークン化されたテキストの親タグとなる “⟨span⟩” では，クリック時に辞書を引き，語義を受け取る動作（図 3(5),(6) にそれぞれ相当）を実現するプログラムが JavaScript で記述され，埋め込まれる．

“PredictGloss” は木構造 $R' \in Dom_T$，注釈関数 g，ユーザ識別子 u，判別器の重み $w^{(k)}$ を受け取り，R' の葉に対して，ユーザ u の非既知語 t を $h(u,t,w^{(k)})$ の符号で判断し，t のみに注釈 $g(t)$ をつけて返す．ただし，R' はトークン化されていると仮定する．注釈関数 g は，トークン $t \in T$ を受け取り，t に注釈をつけた文字列 $g(t)$ をつけて返す関数である．

図 3(5), (6) では，“AJAX”（asynchronous JavaScript and XML）を用いてブラウザとシステムが通信を行う[⋆1]．

⋆1 この通信には，“jQuery” と呼ばれる JavaScript ライブラリを用いている．http://semooh.jp/jquery/

(5) D' 中のトークン t が最初にクリックされると，(4) での予測が訂正されたと判断し，単語の既知・非既知の情報 y をシステムに送出する．(4) で既知と判断されたトークン t がクリックされれば，非既知（$y = 0$）が送出される．(4) で非既知と判断されたトークン t がクリックされれば，既知（$y = 1$）が送出される．

(6) もし $y = 0$，すなわち，非既知の場合は，本システム は t に対応する注釈 $g(t)$ を返し，$g(t)$ がブラウザで表示される．

(u,t,y) のデータの組は，判別器の重みベクトル $\boldsymbol{w}^{(k)}$ を更新するのに使用される．$Update(\boldsymbol{w}^{(k)},(u,t,y))$ の詳細については，§3 で述べる．

§1 で述べたように，本システム には，スケーラビリティと可用性が求められる．本システム は，Web アプリケーションに特化したクラウド計算機環境である Google App Engine (GAE)[⋆2]上で動作するように設計した．GAE は，アップロードされた CGI アプリケーション[⋆3]を運用するサービスであり，ブラウザとの通信にかかる負荷やデータベースへのアクセス負荷がクラウド上で分散させる．また，GAE から Google アカウントを利用することによって，ユーザを識別することが容易になる．

3. 予 測 手 法

項目反応理論[3),4)⋆4]（item response theory, IRT）は，テストの設計に使用される確率的なモデルの総称であり，特に人間の能力を測定するためによく用いられている．TOEFL (Test of English as a Foreign Language) をはじめとする既存の言語テストの設計にも使用されているため，本研究でも項目反応理論を用いることが妥当であると考えられる．項目反応理論を使用した自然言語処理の研究としては，Web から取得した英文テキストから自動的に選択式穴埋め問題を作成する研究が挙げられる[5)]．

項目反応理論は，テスト結果を入力として受け取る．テストは，$|T|$ 個の項目（設問）から構成されるとし，項目の集合を T と書く．被験者の集合を U とし，被験者数を $|U|$ と書く．被験者 $u \in U$ の項目 $t \in T$ に対する反応を $y \in Y$ とすると，(u,t,y) の組が 1 件のテスト結果となる．ただし，Y は，反応の種類の集合である．以上より，テスト結果は，その件数 N 件とすると $\{(u_n,t_n,y_n)|n \in \{1,\ldots,N\}\}$ と表すことが可能である．これが項目反応理論への入力となる．

⋆2 http://code.google.com/intl/ja/appengine/
⋆3 CGI アプリケーションは Python 言語または Java 言語で記述されている．
⋆4 日本語ではその他，**項目応答理論**，**テスト理論**などとも呼ばれる．

項目反応理論は，テスト結果 $\{(u_n, t_n, y_n)|n \in \{1, \ldots, N\}\}$ を入力として受け取り，“パラメータ” を推定する．パラメータは，被験者に関するパラメータと項目に関するパラメータからなる．パラメータを推定することによって，まだ行われていないテストの得点分布を予め予測することや，まだ行われていない被験者と項目の組み合わせに対してその反応を予測することが可能となる．特に，テスト設計の観点からは，前者の応用がよく使われている．一方，本システム は，後者の応用の一種となる．

本システム の応用では，単語クリックログをテスト結果 $\{(u_n, t_n, y_n)|n \in \{1, \ldots, N\}\}$ とみなす．ユーザ $u_n \in U$ の，文書中の個々の単語 $t_n \in T$ に対する反応を $y_n \in Y$ とみなす．Y は，本システム の応用では，$Y = \{0, 1\}$ であるような二値変数とする．$y_n = 1$ のとき，ユーザ u_n は単語 t_n を知っている（既知）とし，$y_n = 0$ のとき，ユーザ u_n は単語 t_n を知らない（非既知）とする．

Y が二値変数であるときの項目反応理論のパラメータについて説明する．ユーザ $u_n \in U$ が単語 $t_n \in T$ を知っている場合，すなわち $y_n = 1$ である確率を，数式 (1) のようにしてモデル化する．

$$P(y_n = 1|u_n, t_n) = c_{t_n} + (1 - c_{t_n})\sigma\left(a_{t_n}\left(\theta_{u_n} - d_{t_n}\right)\right) \tag{1}$$

ここで，σ はロジスティックシグモイド関数である．$x \in \mathcal{R}$ に対し，$x \in \mathcal{R}$, $0 < \sigma(x) < 1$ である．

$$\sigma(x) = \frac{1}{(1 + \exp(-x))} \tag{2}$$

数式 (1) には以下のように 4 つのパラメータがある．項目反応理論は，この 4 つのパラメータを推定する．

θ_{u_n}　被験者 u_n の能力パラメータ．θ_{u_n} が高いほど，被験者 u_n の正答率が増加する．

d_{t_n}　項目 t_n の難易度パラメータ．d_{t_n} が高いほど，被験者 u_n の項目 t_n に対する正答率が低下する．

a_{t_n}　項目 t_n の当て推量パラメータ．$0 \leq c_{t_n} \leq 1$．被験者 u_n が当て推量で項目 t_n に正当する確率を表す．

本研究では，項目反応理論のうち，最も単純な Rasch モデルを用いた．これは，$\forall t \in T$ に対して，$a_t = 1$, $c_t = 1$ の場合である．この場合，数式 (1) は，数式 (3) のようになる．図 3 における $PredictGloss$ の内部で使用される判別関数 h は，$h(u_n, t_n, w^{(k)}) = \log P(y_n = 1|u_n, t_n) - \log P(y_n = 0|u_n, t_n)$ と定義され，$h(u_n, t_n, w^{(k)}) \geq 0$ のとき既知，$h(u_n, t_n, w^{(k)}) < 0$ のとき非既知と判定される．

$$P(y_n = 1|u_n, t_n) = \sigma(\theta_{u_n} - d_{t_n}) \tag{3}$$

Rasch モデルは，本システムで蓄積されるログ $\{(y_n, u_n, t_n)|n \in \{1, \ldots, N\}\}$ を入力として受け取り，ユーザの語彙力 θ_{u_n}，単語の難しさ d_{t_n} のパラメータを最尤推定で計算する．

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{d}} = \mathrm{argmax}_{\boldsymbol{\theta}, \boldsymbol{d}} \prod_{n=1}^{N} P(y_n|u_n, t_n) \tag{4}$$

$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_u, \ldots, \theta_{|U|})$，$\mathbf{d} = (-d_1, \ldots, -d_t, \ldots, -d_{|T|})$ である．

本研究では，語義注釈システムに適応させるために，Rasch モデルとその推定手法をそれぞれ改良した．まず，Rasch モデルの改良について説明する．予測精度を向上させるために，単語の難しさに関する素性を以下のように導入した．$\boldsymbol{e}_u$ を u 番目の要素のみ 1 で他は 0 のサイズ $|U|$ のユニットベクトル，$\boldsymbol{e}_t$ を t 番目の要素のみ 1 で他は 0 のサイズ $|T|$ のユニットベクトルとする．すると，数式 (3) を，重みベクトル $\boldsymbol{w}_{rasch} = (\boldsymbol{\theta} \ \ \boldsymbol{d})^{\mathrm{T}}$ と特徴量ベクトル $\boldsymbol{\phi}_{rasch}(u, t) = (\boldsymbol{e}_u \ \ \boldsymbol{e}_t)^{\mathrm{T}}$ を用いて，数式 (5) と表すことができる．

$$P(y_n = 1|u_n, t_n) = \sigma\left(\theta_{u_n} - d_{t_n}\right) = \sigma(\mathbf{w}_{rasch}^{\mathrm{T}} \boldsymbol{\phi}_{rasch}(u_n, t_n)) \tag{5}$$

数式 (5) において，重みベクトル $\boldsymbol{w}_{rasch}$ を $\boldsymbol{w}_{LR} = (\boldsymbol{\theta} \ \ \boldsymbol{d} \ \ \boldsymbol{w}_a)^{\mathrm{T}}$ に，特徴量ベクトル $\boldsymbol{\phi}_{rasch}$ を $\boldsymbol{\phi}_{LR}(u, t) = (\boldsymbol{e}_u \ \ \boldsymbol{e}_t \ \ \boldsymbol{\phi}_a)^{\mathrm{T}}$ に置き換えることにより，$\boldsymbol{\phi}_a$ に素性を追加することが可能である．追加した素性は，Google 1-gram と SVL12000 である．Google 1-gram は，約 1 兆ページの Web ページ中の英単語の頻度である[6]．SVL12000 は，基本的な語彙 12,000 語に対し，人手で 12 段階の難易度をつけた語彙リストである[7]．

次に，パラメータの推定手法の改良について説明する．パラメータ更新の際に，データセット全体（この場合は，$\{(u_n, t_n, y_n)|n \in \{1, \ldots, N\}\}$）に対して最適化を行うパラメータ推定手法をバッチ学習法という．Rasch モデルのパラメータを最尤推定を用いて推定すると，バッチ学習法となる．対数尤度関数の負をとった $E(\mathbf{w}) = -\log\left(\prod_{n=1}^{N} P(y_n|u_n, t_n; \mathbf{w})\right)$ を定義すると，

$$\begin{aligned} \hat{\boldsymbol{w}} &= \arg\max_{\boldsymbol{w}} \prod_{n=1}^{N} P(y_n|u_n, t_n; \boldsymbol{w}) \\ &= \arg\min_{\boldsymbol{w}} E(\boldsymbol{w}) \end{aligned} \tag{6}$$

となる．実際に，$E(\boldsymbol{w})$ を最小化するために $\nabla E(\mathbf{w})$ を求めると，$-\nabla E(\mathbf{w}) = \sum_{n=1}^{N} -\nabla E_n(\mathbf{w})$ と n に関する和を含む形を含むことから，バッチ学習法であることが分かる．

$Update$ 関数にバッチ学習法を用いると，ユーザがクリックするたびにデータセット全体を参照し最適化を行う必要が生じるので，§1 で述べたスケーラビリティと即応性が低下す

表 1 Dale (1965) の尺度[9] ([10] より抜粋)

Stage 1	“I never saw it before.”
Stage 2	“I’ve heard of it, but I don’t know what it means.”
Stage 3	“I recognize it in context - it has something to do with ...”
Stage 4	“I know it.”

表 2 Paribakht and Wesche (1997) の尺度[11] ([10] より抜粋)

I	“I don’t remember having seen this word before.”
II	“I have seen this word before, but I don’t know what it means.”
III	“I have seen this word before, and I think it means (synonym or translation).”
IV	“I know this word. It means (synonym or translation).”
V	“I can use this word in a sentence: (Write a sentence.) (If you do this section, please also do Section IV)

る．この問題を解決するために，パラメータを逐次的に推定する逐次学習法が提案されている．本システムでは，逐次学習法の 1 つである Stochastic Gradient Descent (SGD)[8] を用いた．SGD では，n に関する和を省略して，$Update$ 関数を次のように定める．

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta_k \nabla E_n(\mathbf{w}^{(k)}) \tag{7}$$

ここで，数式 (7) が収束するように，学習率 η_k は定数 λ, k_0 を用いて $\eta_k = \frac{1}{\lambda(k+k_0)}$ と設定する．SGD は，逐次学習法であり，逐次的にデータが蓄積される本システムに適すると考えられるため，本研究では，SGD を採用した．

4. 実　　験

実験では，テストセットに対する既知/非既知の予測精度を測定した．辞書引きログが N_0 個蓄積されたところに，ユーザが 1 人新規にシステムを使い始め，N_1 個の単語の既知/非既知が得られたと想定し，そのユーザのテストセットに含まれる単語のうち何% について既知/非既知を当てられたかを 1 人の精度とした．

精度評価のために，1 人 12,000 語について，**表 3** に示す 5 段階の自己申告形式で回答させる方法で，被験者（東京大学修士大学院生 16 人）の語彙力を測定した．この語彙力測定の尺度は，Dale (1965) の尺度（**表 1**）と Paribakht and Wesche (1997) の尺度（**表 2**）を参考にした．表 3 のうち，尺度 5 のみを既知の場合とし，残りを非既知の場合とした．12,000 語のうち 11,999 語を $N_1 = 600$ 語までの訓練データセット，1400 語のデベロップメントセット，9999 語のテストセットに分けた．

表 3 5 段階の自己申告形式

1	見たこともない
2	見たことがある気がする
3	確実に見たことはあるが意味は知らない / 覚えたことがあるが意味を忘れている
4	意味を知っている気がする / 意味が推測できる
5	意味を確実に知っている

表 4 追加素性の効果．Rasch が追加素性なし，LR があり．

	$N_1 = 5$	30	100	300	600
IRT	68.33	73.69	74.65	74.65	74.60
LR	**73.25**	**77.89**	**79.09**	**80.03**	**80.01**

単語クリックログの代わりに，単語クリックログと同じデータ構造を持つ **smart.fm** [⋆1] というシステムのログで代用した．**smart.fm** は，Web 上で単語を学習するためのシステムであり，学習済みの単語を学習項目から除外するために既知/非既知をユーザに問う．このデータが，辞書引きログと同じデータ構造を持つ．10,526 人分のデータを **smart.fm** から取得し，継続的にシステムを利用していると思われる 675 人分のデータを実際に用いた．

§3 において，Rasch モデルを拡張した効果を実験を通じて評価した．まず，素性追加の効果について実験を行った．その結果が**表 4**, **図 6** である．“Rasch” が素性を追加していない場合の精度，“LR” が素性を追加した場合の精度である．約 5%精度が向上していることが分かる．

次に，もう一つの改良である逐次化の効果を測定するために，他手法と比較する実験を行った．**図 7** 中の “LR” と “SGD” は，それぞれ，今回使用している Rasch モデルのバッチ学習法[12]，逐次学習法 (SGD) を表す．また，Rasch モデル以外の他の二値分類の手法との比較も行った．図 7，**表 4** 中の “SVM (Linear)” は線形カーネルの SVM (Support Vector Machine)[13]，“SVM (RBF)” は RBF カーネルの SVM，CW は Confidence Weighted という近年提案された逐次学習法[14] であり，今回用いた SGD と比較した[⋆2]．まず，“LR” が $N_1 = 300,600$ の時に，SVM よりも良い精度を達成していることが分かる．この事実は，本システムに Rasch モデルを使用することが，言語テストで実際に使用されている点に加

⋆1 http://smart.fm/
⋆2 CW の実装は次のライブラリを用いた．http://code.google.com/p/oll/wiki/OllMainJa

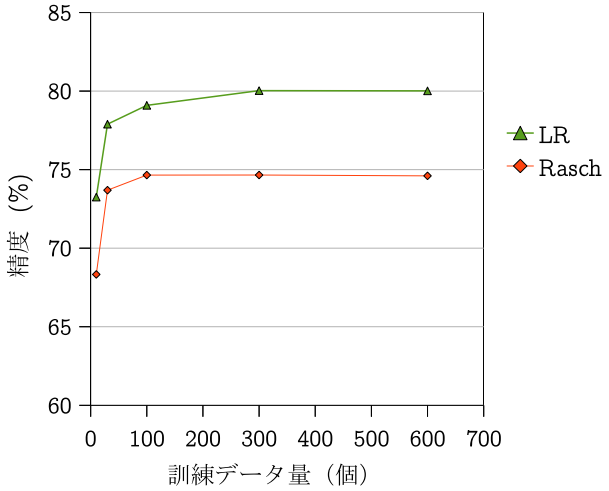

図 6 素性追加の効果

表 5 他手法と比較したときの精度 (%)

	$N_1 = 5$	30	100	300	600
SVM (Linear)	**74.78**	**78.08**	78.88	79.20	79.27
SVM (RBF)	67.61	77.27	**79.16**	79.55	79.91
CW	73.77	72.40	75.06	75.60	75.82
SGD	73,84	73.19	78.50	77.93	78.80
LR	73.25	77.89	79.09	**80.03**	**80.01**

え，予測精度の点からも妥当であることを示している．次に，逐次学習法 “SGD” を用いると，バッチ学習法である “LR” に対して 1%〜2% ほどの精度の減少に抑えられることが分かった．

5. 結論と今後の課題

第二言語で書かれた Web ページを読む際には，ユーザが知らない語（非既知語）が読解の障害となる．この問題に対処するため，ユーザがクリックなどの操作により非既知語の語義を表示することのできる語義注釈システムが提案されてきた．語義注釈システムにおいては，クリックされた単語のログをとることにより，単語クリックログが蓄積される．既存の語義注釈システムでは，単語クリックログが活用されてこなかったが，単語クリックログを解析することにより，読解の障害となる非既知語を予測し，その語に語義の注釈を付与する予測機能を持った語義注釈システムを提案した．

システムの構造では，過去の研究と比較し，使いやすさを考慮して CGI Proxy や AJAX を用いてシステムを設計した．また，高いスケーラビリティと可用性は Google App Engine を用いることにより実現した．

予測には，TOEFL などの言語テストに使用されていることから，項目反応理論の一種である Rasch モデルを使用した．まず，Rasch モデルを改良し，素性を追加して予測精度を向上させた．次に，高いスケーラビリティと即応性を実現するために，Rasch モデルに対してバッチ学習法ではなく，逐次学習法である SGD を用いた．

実験のために，16 人に 1 人 12,000 語について単語を知っている度合いを尋ねた評価用データを作成した．前述の 2 通りの改良について，それぞれ実験を行った．まず，素性追加により判別精度が約 5%向上した．次に，逐次学習法である SGD を使用することによる判別精度の減少は非逐次のバッチ学習法 (LR) に比べて 1〜2% 程度であることが分かった．また，LR は，SVM などの他の 2 値分類手法と比較してもほぼ同程度の精度を達成するこ

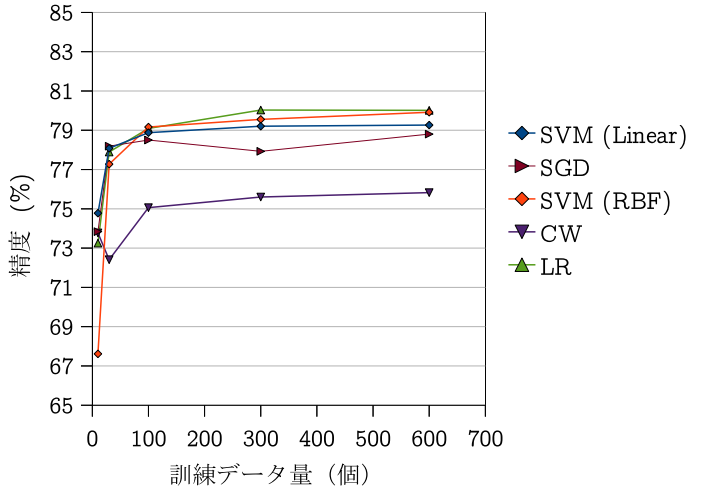

図 7 他手法との比較

とも分かった．

今後の課題としては，ユーザ実験を通じた検証や，ユーザインターフェースの改良，ユーザの英語力で読めそうな Web ページを選別する機能を付与することなどが挙げられる．

参 考 文 献

1) Coolest.com Inc.: popjisyo.com (2002). System available at `http://www.popjisyo.com/`.
2) popIn Inc.: popIn (2008). Software available at `http://www.popin.cc/en/home.html`.
3) 豊田秀樹：項目反応理論 [理論編]-テストの数理-，朝倉書店 (2005).
4) Baker, F. and Kim, S.: *Item response theory: Parameter estimation techniques*, CRC (2004).
5) Sumita, E., Sugaya, F. and Yamamoto, S.: Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions, *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, Ann Arbor, Michigan, Association for Computational Linguistics, pp.61–68 (2005).
6) Brants, T. and Franz, A.: *Web 1T 5-gram Version 1*, Linguistic Data Consortium, Philadelphia (2006).
7) SPACE ALC Inc.: Standard Vocabulary List 12,000 (1998). Data available at `http://www.alc.co.jp/goi/PW_top_all.htm`.
8) Bishop, C.M.: *Pattern recognition and machine learning*, Springer (2006).
9) Dale, E.: Vocabulary measurement: techniques and major findings, *Elementary English*, Vol.42, pp.895–901, 948 (1965).
10) Read, J.: *Assessing Vocabulary*, Cambridge University Press (2000).
11) Paribakht, T. and Wesche, M.: Vocabulary enhancement activities and reading for meaning in second language vocabulary acquisition, *Second language vocabulary acquisition: A rationale for pedagogy*, pp.174–200 (1997).
12) Fan, R., Chang, K., Hsieh, C., Wang, X. and Lin, C.: LIBLINEAR: A library for large linear classification, *The Journal of Machine Learning Research*, Vol.9, pp. 1871–1874 (2008).
13) Chang, C.-C. and Lin, C.-J.: *LIBSVM: a library for support vector machines* (2001). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
14) Dredze, M., Crammer, K. and Pereira, F.: Confidence-weighted linear classification, *ICML '08: Proceedings of the 25th international conference on Machine learning*, New York, NY, USA, ACM, pp.264–271 (2008).